FULL STACK DEVELOPMENT

UNIT – I

SHORT QUESTION AND ANSWERS

1. What is Full Stack Development?

Full Stack Development refers to the process of developing both the frontend and backend parts of a web application. The frontend is responsible for user interface and interaction, while the backend manages server logic, database operations, and APIs. A full stack developer works on both sides of the application. For example, using Angular for frontend, Node.js with Express for backend, and MongoDB for database is a common full stack approach.

2. What are Full Stack Components?

Full stack components are the technologies used to build a complete web application from user interface to database. These components mainly include frontend frameworks like Angular or React, backend technologies like Node.js and Express, and databases like MongoDB. Each component has a specific role, such as displaying UI, handling server logic, and storing data. Together, they provide an end-to-end solution for web development.

3. What is a Web Server?

A web server is a software or system that receives requests from clients through a browser and sends appropriate responses. It processes HTTP or HTTPS requests, executes backend logic, interacts with databases if required, and returns data to the client. For example, a Node.js server using Express can act as a web server that responds to user requests like login or data fetching.

4. What is Browser Rendering?

Browser rendering is the process by which a web browser converts HTML, CSS, and JavaScript code into a visual webpage that users can see and interact with. HTML provides the structure, CSS adds styling, and JavaScript controls dynamic behavior. Browsers like Chrome, Firefox, and Edge use rendering engines to display web pages properly. For example, when a user opens a website, the browser renders text, images, and buttons on the screen.

5. What is Node.js?

Node.js is an open-source JavaScript runtime environment that allows JavaScript to be executed on the server side. It is built on Google Chrome’s V8 engine and supports asynchronous, event-driven programming. Node.js is mainly used to develop fast and scalable backend applications. For example, it is commonly used to create APIs, handle database requests, and manage server operations.

6. What is Angular used for?

Angular is a frontend framework developed by Google and is mainly used to build Single Page Applications. It allows developers to create dynamic and interactive user interfaces using components, data binding, and directives. Angular reduces page reloads and improves performance. For example, in a student portal, Angular can dynamically display marks and profile details without refreshing the page.

7. What is Express.js?

Express.js is a lightweight and flexible web framework built on top of Node.js. It simplifies the process of creating web servers and APIs by providing features like routing, middleware, and request handling. Express is commonly used to manage HTTP requests such as GET and POST. For example, Express can be used to create a REST API for fetching student details from a database.

8. What is MongoDB?

MongoDB is a NoSQL database that stores data in the form of JSON-like documents. Unlike traditional relational databases, it does not use tables and rows. MongoDB is highly scalable and flexible, making it suitable for modern web applications. For example, user details like name, email, and address can be stored as a single document in MongoDB.

9. What is Asynchronous Programming?

Asynchronous programming is a technique where long-running tasks execute in the background without blocking the main program. In Node.js, this allows multiple operations to be handled at the same time. It improves performance and responsiveness of applications. For example, while a file is being read from disk, Node.js can continue handling other user requests.

10. What is an Event Loop?

The Event Loop is a core feature of Node.js that manages asynchronous operations. It continuously checks the event queue and executes callback functions when tasks are completed. The event loop allows Node.js to handle multiple requests using a single thread. For example, database queries and file operations are processed efficiently using the event loop.

11. What is a Callback Function?

A callback function is a function that is passed as an argument to another function and executed later after an operation completes. Callbacks are mainly used in asynchronous programming to handle results of non-blocking tasks. For example, when reading a file, a callback function runs after the file reading is finished.

12. What is NPM?

NPM stands for Node Package Manager and is used to manage external libraries in Node.js projects. It allows developers to install, update, and remove packages easily. NPM also maintains a package.json file that stores project dependencies. For example, the Express framework can be installed using NPM.

13. What is a Package in NPM?

A package in NPM is a reusable piece of code or library that provides specific functionality. Packages help developers avoid writing code from scratch. Examples include express for server creation, mongoose for database connection, and nodemon for automatic server restart.

14. What is package.json?

package.json is a configuration file used in Node.js projects to store project information. It contains details such as project name, version, dependencies, and scripts. This file helps NPM manage required packages automatically. For example, when a project is shared, running npm install installs all dependencies listed in package.json.

15. What is a Single Page Application (SPA)?

A Single Page Application is a web application that loads only one HTML page and updates content dynamically without reloading the page. SPAs provide faster and smoother user experience. Frameworks like Angular and React are used to develop SPAs. For example, Gmail works as an SPA where pages update without refresh.

16. What is JavaScript Event-Driven Programming?

Event-driven programming means executing code based on events such as clicks, requests, or responses. In JavaScript, functions respond to events using callbacks or event listeners. Node.js uses this model to handle server-side events efficiently. For example, when a user sends a request, an event is triggered and handled by the server.

17. What is the role of Backend Services?

Backend services handle the core logic of a web application. They manage authentication, database operations, business rules, and API responses. Backend services run on servers like Node.js. For example, verifying login credentials and fetching user data are backend service operations.

18. What is an API?

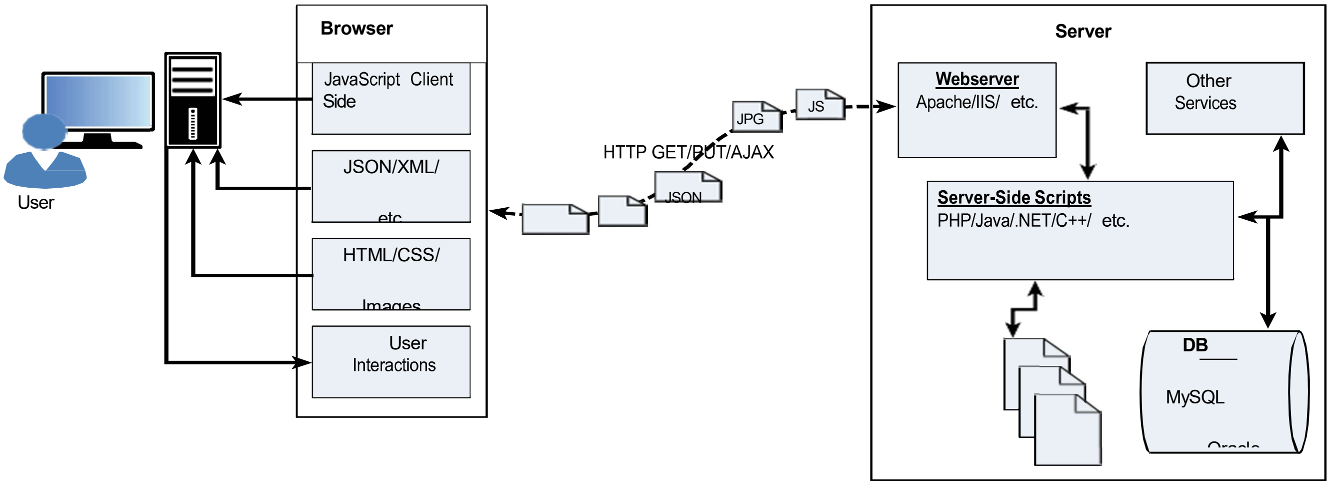
API stands for Application Programming Interface and allows communication between frontend and backend. It defines how requests and responses should be structured. APIs usually use HTTP methods like GET and POST. For example, an API can fetch student details from the server and display them on a web page.

19. What is Routing in a Web Application?

Routing is the process of mapping URLs to specific functions or pages in a web application. It helps the server understand which response to send for a given request. In Node.js, Express is commonly used for routing. For example, accessing /login loads the login functionality.

20. What is the Node.js Event Model?

The Node.js event model is based on handling events using callbacks and the event loop. Instead of blocking execution, Node.js responds to events as they occur. This model improves scalability and performance. For example, multiple users can access a website simultaneously without waiting for each other's requests to complete.



GENERAL QUESTION AND ANSWERS

WEB

The Web (World Wide Web) is a collection of interlinked web pages and resources accessed through the Internet. It allows users to view text, images, videos, and interactive content using a web browser. Web pages are accessed using URLs and HTTP or HTTPS protocols.

HTTP

HTTP (HyperText Transfer Protocol) is a protocol used for communication between a browser and a web server. It defines how requests and responses are exchanged. HTTP is not secure by default.

HTTPS

HTTPS is the secure version of HTTP that uses encryption for data transfer. It protects data from attackers during transmission. HTTPS is commonly used for secure websites like banking and e-commerce.

URL

A URL (Uniform Resource Locator) is the address of a resource on the Internet. It specifies the protocol, domain name, and path to access a webpage or service. For example, https://www.google.com is a URL.

INTERNET

The Internet is a global network that connects millions of computers and servers across the world. It enables communication and data sharing through services like web browsing, email, and file transfer. The Web is one of the services that runs on the Internet.

INTRANET

An Intranet is a private network used within an organization to share information securely. It works similar to the Internet but is accessible only to authorized users. Companies and institutions use intranets for internal communication and resources.

STACK

A stack refers to a set of technologies used together to build a complete application. It usually includes frontend, backend, and database tools. For example, the MEAN stack uses MongoDB, Express, Angular, and Node.js.

FRAMEWORK

A framework is a structured platform that provides predefined libraries, tools, and rules for application development. It helps developers build applications faster and with better organization. Examples include Angular, Express, and Django.

IDE

An IDE (Integrated Development Environment) is a software tool that provides code editor, compiler, debugger, and other development tools in one place. It helps programmers write, test, and debug code easily. Examples include VS Code and Eclipse.

SERVER

A server is a computer or software that provides services or data to client systems. It processes requests from clients and sends responses over a network. Servers are used to host websites, applications, and databases.

WEB SERVER

A web server is a server that handles HTTP and HTTPS requests from browsers. It delivers web pages, images, and data to users. Examples of web servers include Node.js, Apache, and Nginx.

BROWSER ROLES

A browser is used to request and display web content from web servers. It renders HTML pages, applies CSS styles, and executes JavaScript code. Browsers also allow users to interact with websites using forms and links.

API

An API (Application Programming Interface) allows different software applications to communicate with each other. It defines how data is requested and sent between systems. APIs are commonly used to connect frontend and backend components.

JSON

JSON (JavaScript Object Notation) is a lightweight data format used to exchange data between systems. It is easy to read and write for both humans and machines. JSON is widely used in APIs and web applications.

XML

XML (Extensible Markup Language) is a markup language used to store and transport structured data. It uses tags to define data elements. XML is commonly used in configuration files and data exchange.

DIFFERENCE BETWEEN WEBSITE AND WEB APPLICATION

A website mainly provides static information for users to read, such as blogs or news pages. A web application is interactive and allows users to perform actions like login, data entry, and transactions. Web applications involve more backend processing than websites.

NODE.JS

Node.js is a JavaScript runtime environment used to run JavaScript on the server side. It supports asynchronous and event-driven programming. Node.js is widely used for building fast and scalable backend applications.

ANGULAR

Angular is a frontend framework developed by Google for building single page applications. It uses TypeScript and follows a component-based architecture. Angular helps in creating dynamic and responsive user interfaces.

REACT

React is a JavaScript library used for building user interfaces, especially single page applications. It uses a component-based approach and virtual DOM for better performance. React is maintained by Facebook.

MONGODB

MongoDB is a NoSQL database that stores data in JSON-like document format. It is flexible, scalable, and suitable for large and dynamic data. MongoDB is commonly used in modern web applications.

FRONTEND

Frontend is the client-side part of a web application that users interact with directly. It includes the design and layout of web pages using HTML, CSS, and JavaScript. Frameworks like Angular and React are used in frontend development.

BACKEND

Backend is the server-side part of a web application that handles logic, database operations, and authentication. It processes client requests and sends responses. Technologies like Node.js and Express are used for backend development.

DATABASE LAYER

The database layer is responsible for storing and managing application data. It ensures data consistency, security, and fast retrieval. Databases like MongoDB, MySQL, and SQLite are used in this layer.

SQLITE

SQLite is a lightweight relational database that stores data in a single file. It does not require a separate server and is easy to use. SQLite is commonly used in mobile and small-scale applications.

CSS

CSS (Cascading Style Sheets) is used to style and design web pages. It controls layout, colors, fonts, and responsiveness. CSS improves the visual appearance of websites.

BOOTSTRAP

Bootstrap is a CSS framework used to create responsive and mobile-friendly web pages. It provides ready-made components like buttons, forms, and grids. Bootstrap reduces development time.

STREAMLIT

Streamlit is a Python framework used to create interactive web applications for data science and machine learning. It allows developers to build dashboards quickly with minimal code. Streamlit is mainly used for data visualization.

EXPRESS

Express (Express.js) is a lightweight and flexible web application framework built on top of Node.js. It is used to develop web servers and RESTful APIs easily by simplifying routing, request handling, and response management. Express supports middleware, which helps in tasks like authentication and logging. It is widely used in backend development for building fast and scalable web applications.